Base de données multidimensionnelles implantée en NoSQL

Membres :

Arnauld ALEX

Nicolas AUBERT

Kristen VIGUIER

Client : Mr.TESTE

Superviseur : Mr.ROQUES

Dans le cadre du PTUT à l'IUT de Blagnac 2014-2015

Accroche

En entreprise :

- Créer une base de données
- Modélisation en schéma
- Génération d'un script d'implémentation

Générateur de script à partir d'un schéma :

- Permettre de gagner du temps (grosses bases de données ex: multidimensionnelles)
- Choisir le format
- Fichier exploitable

Problématique

Comment pouvons-nous assister l'utilisateur pour concevoir un schéma de base de données multidimensionnelles et lui permettre de générer un script de création du schéma en NoSQL ?

PLAN

- DESCRIPTION DU PROJET
- EMF - ACCELEO
- TRADUCTEUR
- MONGODB
- PARTIE TECHNIQUE
- BILAN

DESCRIPTION DU PROJET

Qu’est-ce qu’une base de données multidimensionnelles ?

- Faire face au besoin de stocker plus de données
- Y accéder plus rapidement et plus pertinemment
- Permettre l’aide à la décision
- Être dotée de plus de dimensions que le modèle relationnel

DESCRIPTION DU PROJET

Objectifs demandés par le client :

- Créer une IHM de conception de schéma de base de données multidimensionnelles
- Créer un module de traduction automatique pour produire un script de création de la base
- Le script doit être utilisable dans un SGBD NoSQL nommé MongoDB.
- Etudier les fonctionnalités d'installation et d'administration du système MongoDB

IHM : Interface Homme Machine

SGBD: Système de Gestion de Base de Données

EMF - ACCELEO

EMF - ACCELEO

EMF (Eclipse Modeling Framework) :

- Framework de modélisation
- Infrastructure de génération de code
- Basé sur des modèles de données structurées

- Produire des classes Java (représentant le modèle)
- Visualiser les classes
- Les éditer avec un système de commandes
- Les manipuler dans un éditeur.

EMF - ACCELEO

Acceleo :

- Générateur de codes
- Permettre de réaliser des applications
- Génération de codes depuis des modèles.
- Réaliser des générations incrémentales.

EMF - ACCELEO

- EMF permet de créer un méta-modèle (Ecore)
- Ecore génère du code Java
- Compiler les classes dans un plugin
- Plugin -> permet de créer un modèle de BD en fonction du méta-model

Modèle

- Acceleo génère du code qui implémente des fonctions : parcourir/traiter le modèle de base créé par l'utilisateur
- Génération d'un patron exploitable avec le traducteur

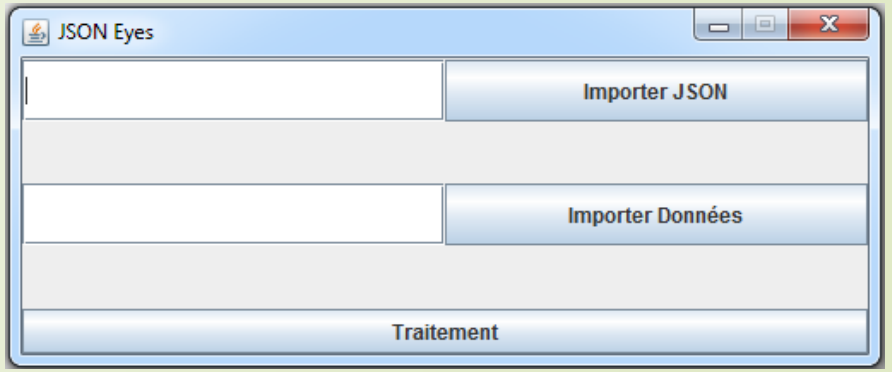
```
{"id":XXXXXX,"A":{"nombre":XXXXXX}}
```

Patron

TRADUCTEUR

TRADUCTEUR

- Entrée :
  - Un patron format ‘Json’
  - Un fichier de données Excel au format ‘CSV’
- Sortie :
  - Un fichier ‘Json’ rempli avec les données

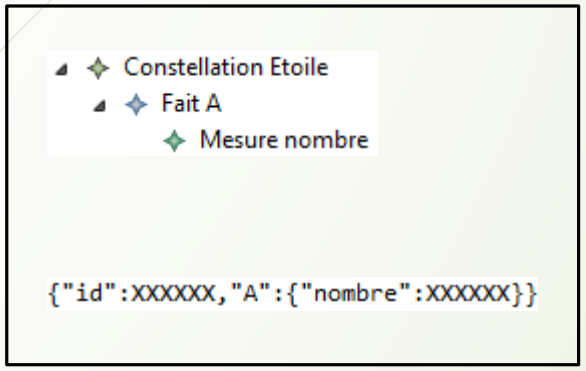


TRADUCTEUR

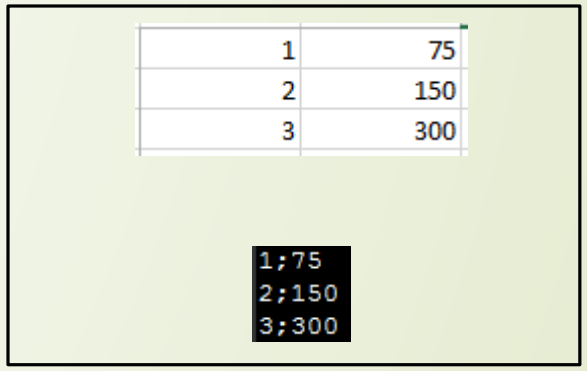
- Fonctionnement :
  - Découpage du patron
  - Remplacement des caractères XXXXXX par les données
  - Réassemblage de la ligne
  - Stockage de la ligne réassemblée

  - Une fois que toutes les données ont été extraites et traitées :
    - Écriture des lignes dans le fichier ‘Json’

TRADUCTEUR

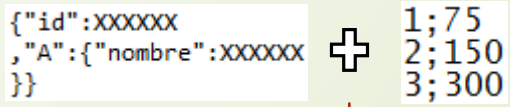


Patron

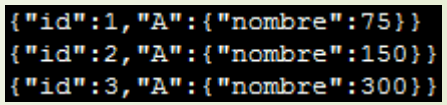


Données

Entrée



Traitement

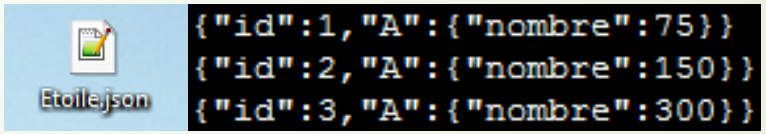


JSON Exploitable

Sortie

TRADUCTEUR

- Résultat :
  - Fichier ‘JSON’ exploitable par MongoDB



MongoDB

MongoDB

- Système de Gestion de Base de Données :
  - Répartissable sur plusieurs ordinateurs
  - Pas de schéma relationnel précis
  - Importation facile de fichiers JSON

- Dans le cadre du projet
  - Importation du fichier venant du traducteur pour exploiter les données

- Pour ce faire
  - Commande fournie par mongoDB (mongoimport)

MongoDB

- Nouveautés :
  - Installation du logiciel en monoposte (guide d'installation)
  - Commandes nouvelles et complexes (Manipulation des données en JavaScript)
  - Importation de fichiers au format JSON et/ou csv mais pas uniquement SQL
  - Travail sur invite de commandes

```
C:\Users\Nico\Documents\Cours\PTUT>mongo ConstellationVente
MongoDB shell version: 2.6.7
connecting to: ConstellationVente
>
```

MongoDB

- Problèmes rencontrés
  - Nouveau système avec normes différentes d'Oracle
  - Difficultés sur les importations/exportations de données
  - Mise en norme avec le traducteur

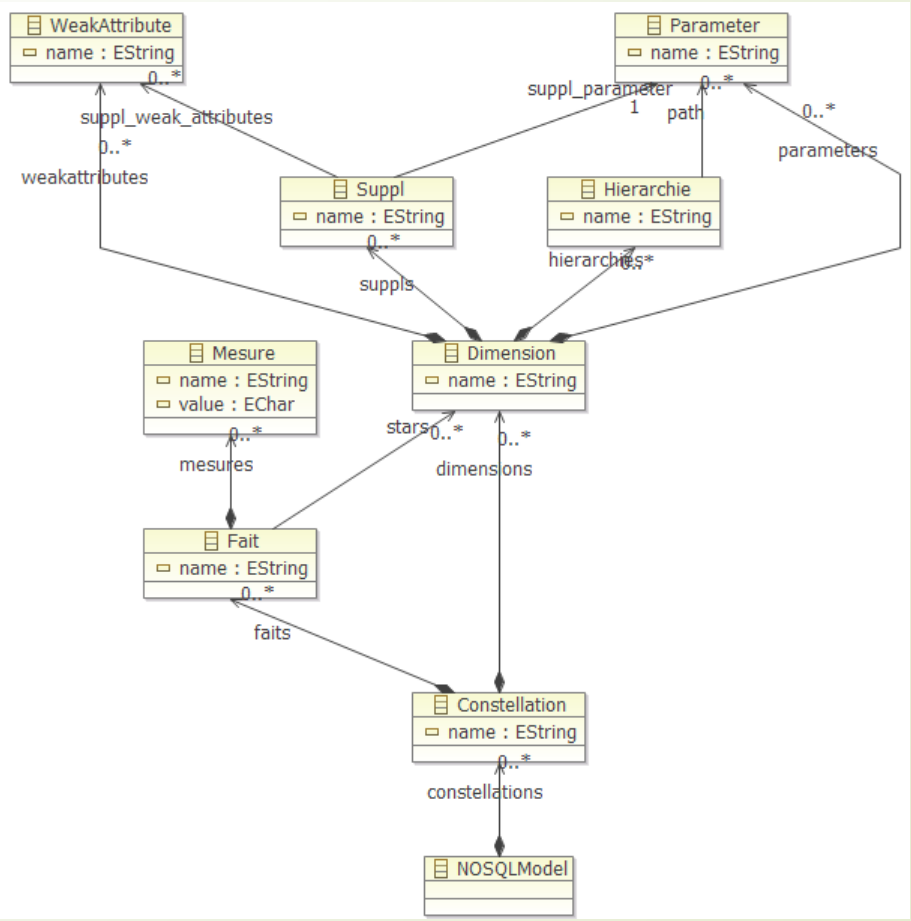
- Solutions
  - Recherches de documentations sur MongoDB
  - Créations de plusieurs fichiers JSON (format)
  - Mise en commun avec le fichier du traducteur

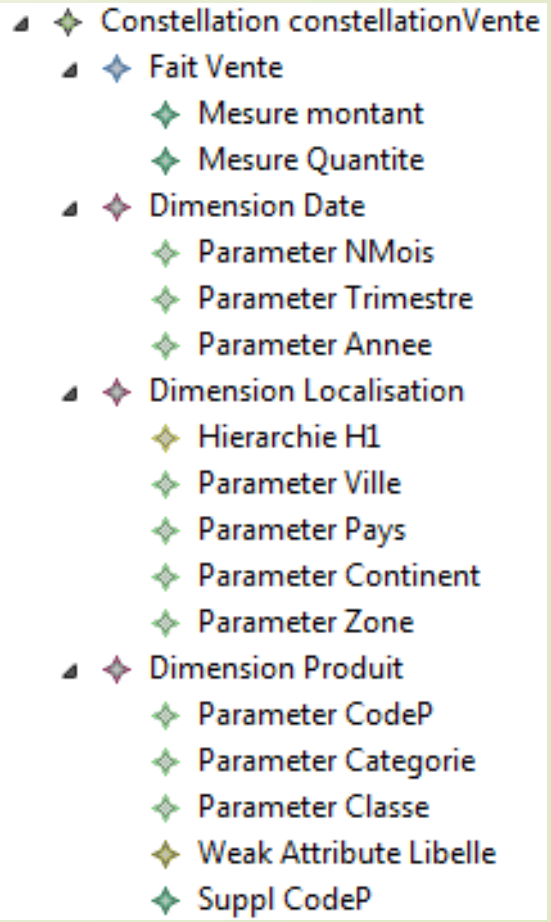
```
C:\Users\Nico\Documents\Cours\PTUT>mongoimport -type json -d ConstellationVente
-c ConstellationVente --file constellationVente.json
connected to: 127.0.0.1
2015-01-30T12:05:44.111+0100 check 9 20
2015-01-30T12:05:44.115+0100 imported 20 objects
```

PARTIE TECHNIQUE
retour sur le méta-modèle EMF

PARTIE TECHNIQUE

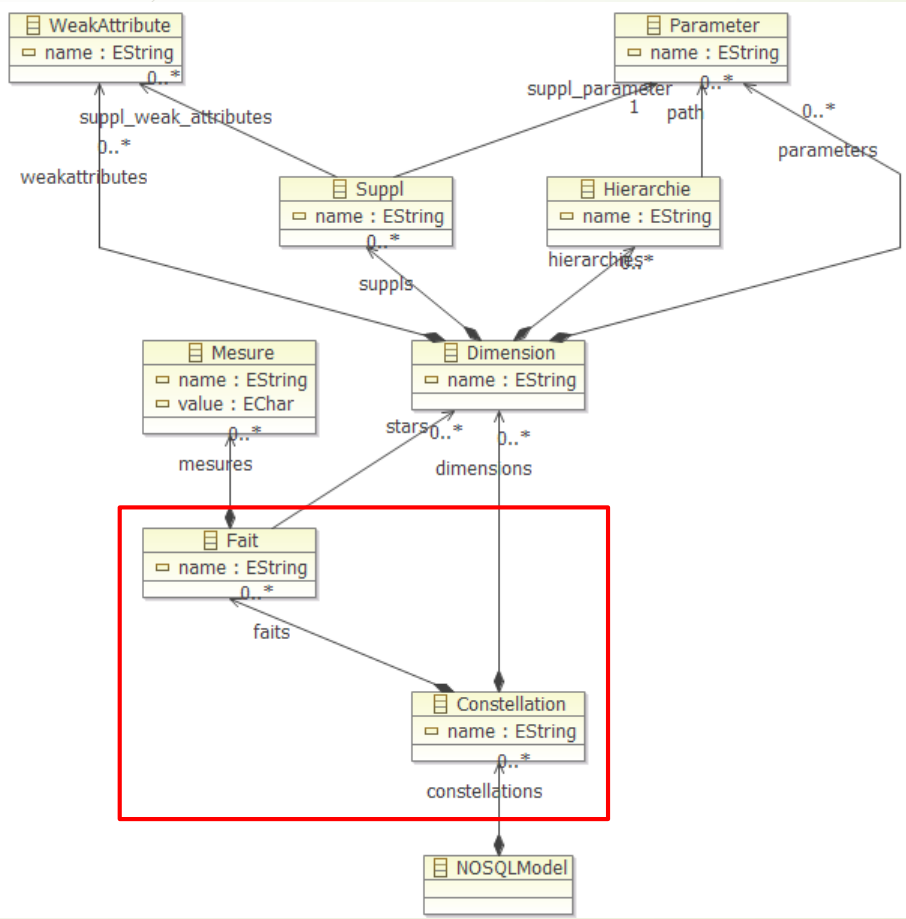
EMF





Gauche : méta-model
Droite : modèle d'une base

PARTIE TECHNIQUE



Exemple de lecture :

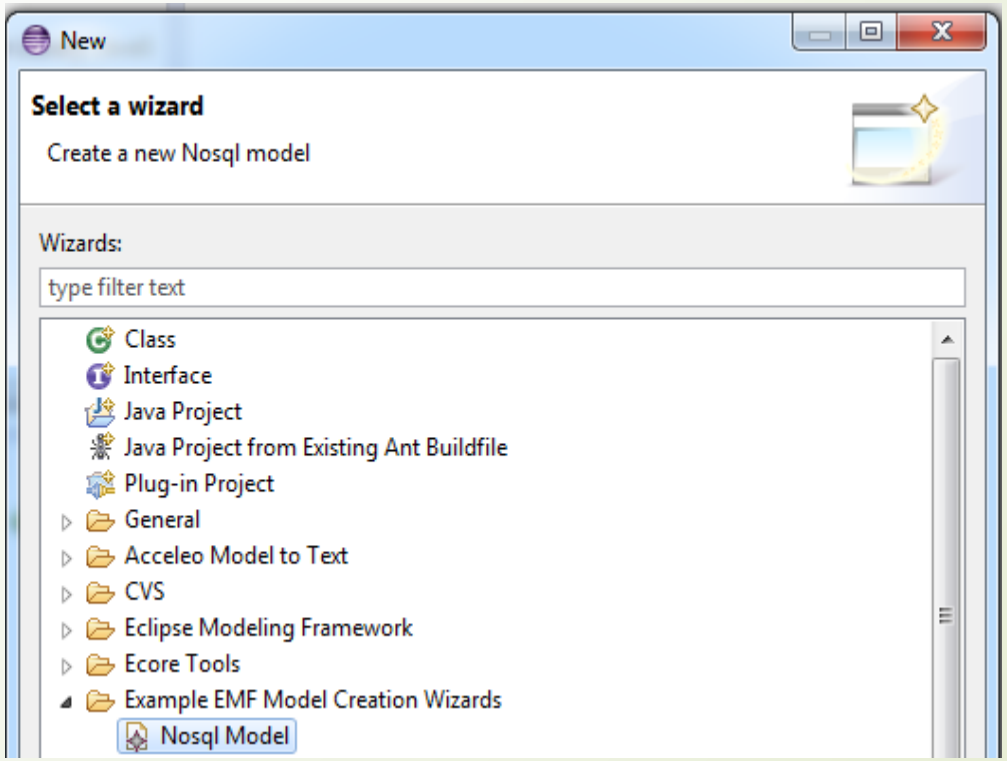
Une constellation se compose de :
- aucun ou plusieurs faits
- aucune ou plusieurs dimensions

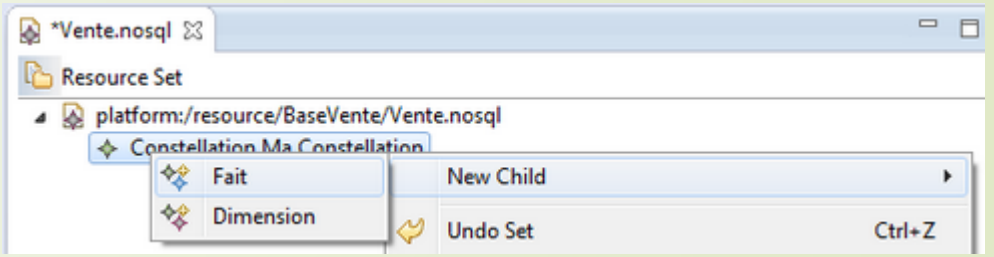
De plus, dans un fait, il peut y avoir plusieurs dimensions qui lui appartiennent

Une constellation a un nom de type String
Un fait a un nom de type String

PARTIE TECHNIQUE

Déploiement en extension de logiciel (plugins) :

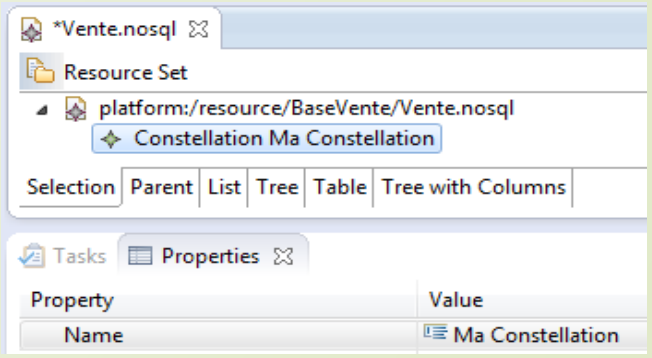




Apparition de nouveaux menus :

grâce au méta-model

grâce aux classes générées par l'outil



BILAN

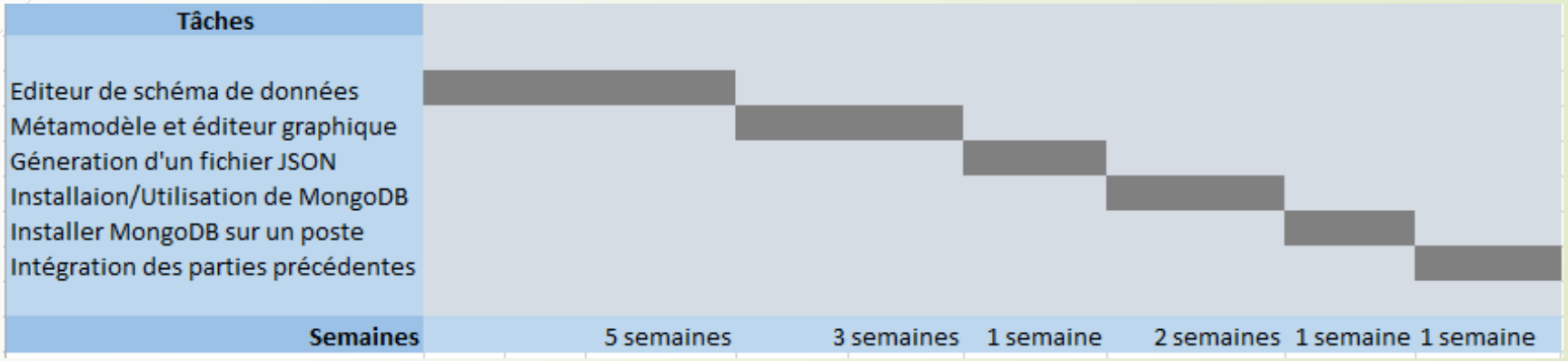
BILAN

Organisation et mise en place du projet

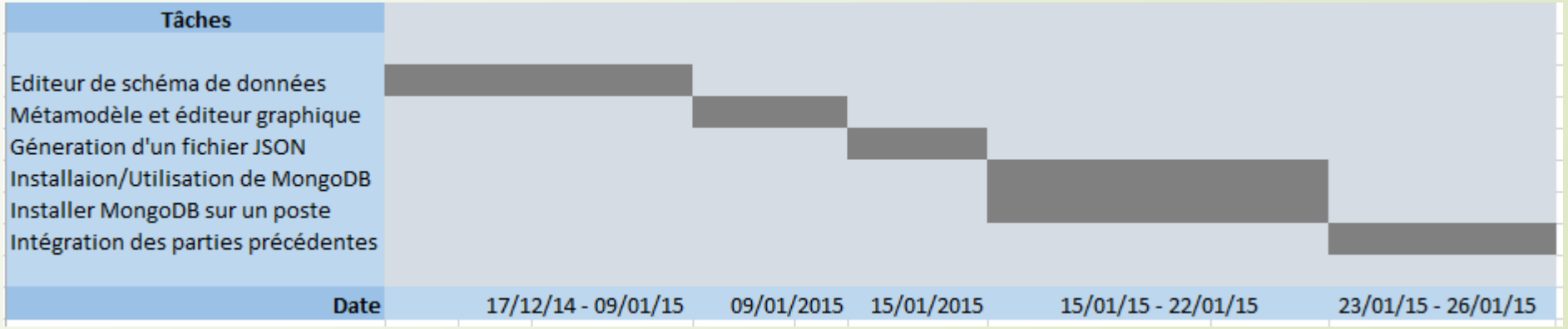
- Consultation du client pour ses besoins
- Deux solutions proposées par le superviseur :
  - Soit coder nous-même (long et compliqué)
  - Soit utiliser des méta-modèles et un générateur de texte. (EMF / Acceleo)
- Opter pour la deuxième solution :
  - plus pratique
  - découverte d'une nouvelle technologie complexe

BILAN

Prévisionnel :



Réel :



BILAN

Déroulement du projet

- Perte d'un membre (-)
- Difficulté dans la gestion du retard (-)
- Manque de communication avec le client (-)
- Complexité des nouvelles technologies (-)

- Adaptation (+)
- Travail sous pression (+)
- Technologies intéressantes (+)

Améliorations pour de futurs projets :

- Rendez-vous plus fréquent avec le client
- Constance dans le travail
- Respect du Gantt
- Décision d'une méthode de projet

BILAN

Travail fourni :

- Plusieurs plugins Java pour Eclipse
  - Permettre de créer le modèle de la base de données
  - Générer un fichier JSON patron à partir du schéma
- Un traducteur (Patron -> JSON complet)
  - Permettre au traducteur d'affecter les valeurs au fichier JSON grâce à un fichier de données CSV
- Manuel pour déployer le fichier complet sur MongoDB

BILAN

Quantité de travail

- Quatre plugins NoSQL pour la création du schéma + Documentations
  - 39 classes générées (12900 lignes)
- Deux plugins Acceleo pour la génération du patron + Documentations
- Une application Java ‘Traducteur’ + Documentations
  - 5 classes créées (406 lignes)
- Manuel d’installation MongoDB + Documentations

CONCLUSION

Résultat obtenu :

- Un projet Eclipse pour le méta-modèle
- Un traducteur Java avec fichiers de tests
- Un manuel d'installation

Améliorations possibles :

- Cluster
- BigData
- Intégration du traducteur dans Eclipse
- Modifications mineures

Contact :

- kristen.viguier@gmail.com
- arnauld.alex@gmail.com
- aubertnicolas9@gmail.com